NASA Contractor Report 201629

ICASE Report No. 96-72

ICASE

DISCRETE DETERMINISTIC AND STOCHASTIC PETRI NETS

Robert Zijal
Gianfranco Ciardo

NASA Contract No. NAS1-19480
December 1996

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA 23681-0001

Operated by Universities Space Research Association

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

Discrete Deterministic and Stochastic Petri Nets

Robert Zijal*

Technische Universität Berlin,
Institut für Technische Informatik,
Franklinstr. 28/29, 10587 Berlin, Germany
bob@cs.tu-berlin.de

Gianfranco Ciardo†

Department of Computer Science
College of William and Mary
Williamsburg, VA 23187-8795, USA
ciardo@cs.wm.edu

Abstract

Petri nets augmented with timing specifications gained a wide acceptance in the area of performance and reliability evaluation of complex systems exhibiting concurrency, synchronization, and conflicts. The state space of time-extended Petri nets is mapped onto its basic underlying stochastic process, which can be shown to be Markovian under the assumption of exponentially distributed firing times. The integration of exponentially and non-exponentially distributed timing is still one of the major problems for the analysis and was first attacked for continuous time Petri nets at the cost of structural or analytical restrictions. We propose a *discrete deterministic and stochastic Petri net* (DDSPN) formalism with no imposed structural or analytical restrictions where transitions can fire either in zero time or according to arbitrary firing times that can be represented as the time to absorption in a finite absorbing *discrete time Markov chain* (DTMC). Exponentially distributed firing times are then approximated arbitrarily well by geometric distributions. Deterministic firing times are a special case of the geometric distribution. The underlying stochastic process of a DDSPN is then also a DTMC, from which the transient and stationary solution can be obtained by standard techniques. A comprehensive algorithm and some state space reduction techniques for the analysis of DDSPNs are presented comprising the automatic detection of conflicts and confusions, which removes a major obstacle for the analysis of discrete time models.

*The research of the first author was supported by the German Research Council (DFG) under grant Ho 1257/7-1.

†This research was supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the second author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

1 Introduction

Petri nets (PN) [13] proved to be a powerful graphical and mathematical modeling tool that allows to describe and analyze complex systems exhibiting concurrency, synchronization, and conflicts. The ability to model timed and probabilistic behavior is essential in the field of performance and reliability evaluation. This need leads to various different extensions of the PN formalism, where the class of *stochastic Petri nets* (SPNs) gained the widest acceptance. In SPNs, firing time delays are specified by probability distributions associated to transitions. SPNs are often classified as continuous or discrete time, depending on the type of firing time distributions and on the underlying stochastic process.

Deterministic and stochastic Petri nets (DSPNs) [3] represent the most important continuous time approach where transitions can fire either in zero time or after a constant (deterministic) or exponentially distributed time delay. The initial definition of DSPNs imposed the structural restriction that concurrent deterministic activities cannot be present. This problem was theoretically solved in [10]. However, the solution is not feasible in practice because it leads to a state space explosion when a larger number of concurrent deterministic activities is to be considered.

Discrete time stochastic Petri nets [12] belong instead to the discrete approach, where transition firing times are specified by geometric distributions which approximate the exponential distribution arbitrarily well in discrete time. Other approaches having an underlying discrete time stochastic process have been presented in [15] (Timed Petri nets) and in [11] (Generalized Timed Petri nets), but they does not achieve the modeling power of DSPNs.

The mixture of deterministic and stochastic firing times still imposes severe problems on the quantitative analysis of a time-extended PN, since the state space needs to be generated and mapped onto the basic underlying stochastic process. Our work attacks this problem by adopting a pure discrete-time approach. However, conflicts and confusions among transition firings are more likely to occur in discrete than in continuous time, since transitions are allowed to fire only at certain discrete instants of time. Thus, simultaneous firing attempts of all transitions, including the timed transitions, can take place. The detection of the sets of transitions involved in conflicts and confusions is a precondition for the correct specification of probabilistic firing weights resolving these situations. This is an important and oftenly neglected issue especially for discrete time models.

In [14], *Discrete time Deterministic and Stochastic Petri nets* (dtDSPNs) were introduced where transitions fire either in zero time or after a constant or geometrically distributed time delay without any structural restriction. The deterministic time delay is then modeled as a special case of the geometric distribution. In dtDSPNs, the problem of conflicts and confusions is relaxed to a certain degree by an unconventional approach. The sequentialization of simultaneously fireable timed tran-

sitions is not enforced, which leads to the elimination of confusion situations for timed transitions. The drawback of this approach is that a dtDSPN model can generate states which are not covered by the classical Petri net theory.

A more general approach was proposed in [5] with *Discrete Time Markovian SPNs* (DTMSPNs), where firing time distributions are specified by arbitrary finite absorbing DTMCs. It has been proven in [5] that the underlying stochastic process of a DTMSPN is a DTMC, provided that the modeler detects and resolves all conflicts and confusions manually, possibly a very difficult task. This drawback lead in [8] to the development of a new method for the automatic detection of conflicts and confusions applicable to all types of stochastic Petri nets. This approach is independent of structural PN properties and is solely based on the state space generation of a given model, so that only actually occurring conflicts and confusions are detected. This is is not the case for the structural tests employed in continuous time approaches, which are based on necessary, not sufficient, conditions. Thus, structural tests can lead to an overspecification of a given model resulting in a more difficult correct interpretation of obtained results measures.

The work presented in this paper combines the results of [14], [5], and [8], while removing the mentioned drawbacks of [14] and [5]. We define *discrete deterministic and stochastic Petri nets* (DDSPNs). In DDSPNs, transitions can fire either in zero time or after a time delay specified by arbitrary finite absorbing DTMCs without any structural restriction. Firing time distributions of a DDSPN include the geometric and the deterministic distribution as a special case. Any other discrete distribution that can be expressed by a finite absorbing DTMC can be freely defined, such as the discrete uniform distribution. We adapt the general approach for the automatic detection of conflicts and confusions from [8] and integrate it into the solution method for the analysis of DDSPNs. Together with the solution method, a new algorithm for the complex and non-trivial state-space generation is presented, mapping a DDSPN onto a DTMC, from which again the transient and stationary solution can be obtained by standard techniques. Finally, some state space reduction techniques for DDSPNs are proposed to relax the inherent problem of state space explosion.

Section 2 defines untimed PNs. Section 3 introduces the discrete firing time distributions of DDSPN transitions. Sections 4 and 5 present the complete DDSPN formalism itself and the corresponding state space reduction methods. Numerical results are shown in Section 6, followed by concluding remarks in Section 7.

2 The PN Formalism

We recall the (extended) PN formalism used in [8]. See also [4] for more details on PNs with marking-dependent arc multiplicities. A PN is a tuple $\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}\right)$ where:

- P is a finite set of *places*, which can contain tokens. A marking $\mu \in \mathbb{N}^{|P|}$ defines the number of tokens in each place $p \in P$, indicated by μ_p (when relevant, a marking should be considered a column vector). D^-, D^+, D°, and g are "marking-dependent", that is, they are specified as functions of the marking.
- T is a finite set of *transitions*. $P \cap T = \emptyset$.
- $\forall p \in P, \forall t \in T, \forall \mu \in \mathbb{N}^{|P|}$, $D^-_{p,t}(\mu) \in \mathbb{N}$, $D^+_{p,t}(\mu) \in \mathbb{N}$, and $D^\circ_{p,t}(\mu) \in \mathbb{N}$ are the multiplicities of the *input arc* from p to t, the *output arc* from t to p, and the *inhibitor arc* from p to t, when the marking is μ, respectively.
- $\succ \subseteq T \times T$ is an acyclic (*pre-selection*) *priority relation*.
- $\forall t \in T, \forall \mu \in \mathbb{N}^{|P|}, g_t(\mu) \in \{0, 1\}$ is the *guard* for t in marking μ.
- $\mu^{[0]} \in \mathbb{N}^{|P|}$ is the *initial marking*.

Places and transitions are drawn as circles and rectangles, respectively. The number of tokens in a place is written inside the place itself (default is zero). Input and output arcs have an arrowhead on their destination, inhibitor arcs have a small circle. The multiplicity is written on the arc (default is the constant 1); a missing arc indicates that the multiplicity is the constant 0. The default value for guards is the constant 1.

Let $\mathcal{E}(\mu)$ be the set of transitions *enabled* in marking μ. A transition $t \in T$ is enabled in marking μ if, and only if, its guard evaluates to 1, its input and inhibitor arc conditions are satisfied, and no other transition with pre-selection priority over t is enabled (this is well defined because $\succ$ is acyclic):

$$(g_t(\mu) = 1) \wedge \left(\forall p \in P, D^-_{p,t}(\mu) \leq \mu_p \wedge \left(D^\circ_{p,t}(\mu) > \mu_p \vee D^\circ_{p,t}(\mu) = 0\right)\right) \wedge (\forall u \in \mathcal{E}(\mu), u \not\succ t).$$

A transition $t \in \mathcal{E}(\mu)$ can fire, causing a change to marking $\mathcal{M}(t, \mu)$, obtained from μ by subtracting the *input bag* $D^-_{\bullet,t}(\mu)$ and adding the *output bag* $D^+_{\bullet,t}(\mu)$ to it:

$$\mathcal{M}(t, \mu) = \mu - D^-_{\bullet,t}(\mu) + D^+_{\bullet,t}(\mu) = \mu + D_{\bullet,t}(\mu),$$

where $D = D^+ - D^-$ is the incidence matrix. $\mathcal{M}$ can be extended to its reflexive and transitive closure by considering the marking reached from μ after firing a sequence of transitions. The *reachability set* is then given by

$$\mathcal{S} = \{\mu : \exists \sigma \in T^* \wedge \mu = \mathcal{M}(\sigma, \mu^{[0]})\},$$

where T^* indicates the set of transition sequences. The *reachability graph* is $(\mathcal{S}, \mathcal{A})$, where $\mathcal{A}$ contains an arc $\mu \xrightarrow{t} \mu'$ iff $\mu \in \mathcal{S}$, $t \in T$, and $\mu' = \mathcal{M}(t, \mu)$.

3 Discrete Time Phase Distributions

Firing times of transitions in DDSPNs are modeled by *discrete time phase distributions* (DTPs).

Definition 3.1 A DTP is represented by a finite absorbing *discrete time Markov chain* (DTMC) $\{X_{i\delta} | i \in \mathbb{N}\}$ where

- $\delta > 0$ is the underlying constant time-step.
- $X_{i\delta} \in I = \{n, n-1, ..., 0\}$, the finite state space of the DTMC. Each state corresponds to a possible distribution of the *remaining firing time* (RFT) for a transition.
- $\forall k \in I, \Pr\{X_0 = k\}$ is the initial probability distribution, such that $\sum_{k \in I} \Pr\{X_0 = k\} = 1$.

States $I \setminus \{0\}$ are transient. State 0 is absorbing and represents the case that a phase reached zero and that the corresponding transition is allowed to fire. □

Two additional symbolic DTP states, whose sojourn times are zero, b and $\bullet$, are introduced. The symbolic state b (for *begin*) represents the initial probability distribution of a DTP. Since DTPs will be used for modeling RFTs of transitions, a second symbolic state $\bullet$ is needed to represent unambiguously the case when a transition is disabled and no definite phase is specified.

Special cases of a DTP are for instance the geometric, constant, and the uniform distribution. In the following we will show how these discrete distributions can be represented by DTPs. From now on, the states of I will be referred to as *phases*, to make a clear distinction between DTP states and the overall state space of a DDSPN.

Geometric Distribution

The geometric distribution, $\text{Geom}(\alpha, \omega)$ with probability $\alpha \in (0, 1)$, approximates the exponential distribution in discrete time arbitrarily well as its unit-step $\omega > 0$ decreases. The *probability mass*

function (pmf), *cumulative probability distribution function* (CDF), and expectation of a random variable $X \sim \text{Geom}(\alpha, \omega)$ are then given by

- pmf: $p_X(i\omega) = \begin{cases} \alpha(1-\alpha)^{i-1} & \text{if } i \in \mathbb{N}^+, \text{ where } \mathbb{N}^+ = \{1, 2, 3, ...\}, \\ 0 & \text{otherwise.} \end{cases}$
- CDF: $F_X(x) = \begin{cases} 1 - (1-\alpha)^{\lfloor \frac{x}{\omega} \rfloor} & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$
- mean: $\frac{\omega}{\alpha}$ (average delay)

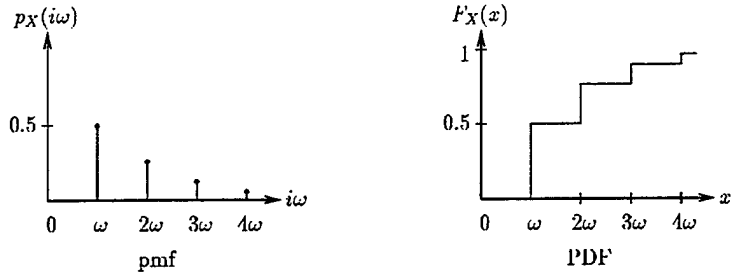


Figure 1: $X \sim \text{Geom}(0.5, \omega)$.

Fig. 1 shows the pmf and CDF of $X \sim \text{Geom}(0.5, \omega)$. The DTP representation of $\text{Geom}(\alpha, \omega)$ depends on its unit-step ω, defined as an arbitrary non-negative integer multiple, $\omega = c\delta, c \in \mathbb{N}^+$, of the constant basic underlying time-step $\delta > 0$ of all DTPs. For example, two geometric distributions, $\text{Geom}(1/5, \delta)$ and $\text{Geom}(4/5, 4\delta)$, have different DTPs but the same mean value of 5δ. Fig. 2 shows the DTP representations of $X \sim \text{Geom}(\alpha, \delta)$ and $X \sim \text{Geom}(\alpha, 4\delta)$. The states of a DTP (or phases) are represented by nodes and the probabilistic state transitions by labeled arcs.

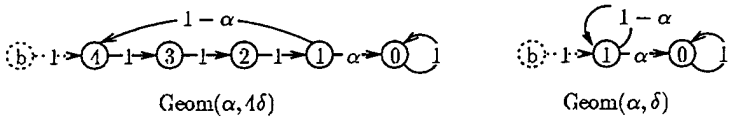


Figure 2: DTP representations of geometric distributions.

Constant Distribution

The constant distribution, $\text{Const}(\omega)$ with $\omega > 0$, can be seen as a special case of the geometric distribution, where $\alpha = 1$, so $\text{Const}(\omega) = \text{Geom}(1, \omega)$. The pmf and CDF (Fig. 3) of a random variable $X \sim \text{Const}(\omega)$ are then given by

- pmf: $p_X(i\omega) = \begin{cases} 1 & \text{if } i = 1, \\ 0 & \text{otherwise.} \end{cases}$
- CDF: $F_X(x) = \begin{cases} 0 & \text{if } x < \omega, \\ 1 & \text{otherwise.} \end{cases}$

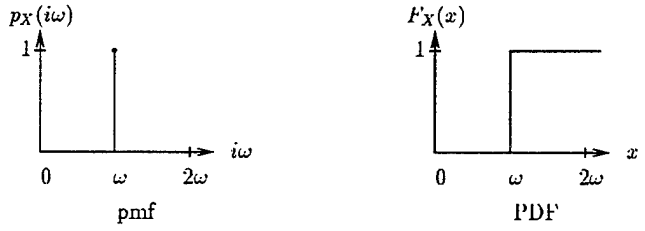


Figure 3: $X \sim \text{Const}(\omega)$.

Immediate transitions (firing in zero time) can be modeled by a special case of the constant distribution where $X \sim \text{Const}(0)$. Fig. 4 shows the DTP representations of $X \sim \text{Const}(0)$ and $X \sim \text{Const}(4\delta)$.

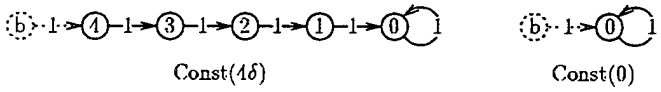


Figure 4: DTP representations of constant distributions.

Discrete Uniform Distribution

Like the geometric distribution, the discrete uniform distribution, $\text{Unif}(a, b, \omega)$ where $a, b \in \mathbb{N}$, $a < b$, and ω is a multiple of δ. The pmf, CDF, and expectation of a random variable $X \sim \text{Unif}(a, b, \omega)$ are then given by

- pmf: $p_X(i\omega) = \begin{cases} (b-a+1)^{-1} & \text{if } i \in \{a, a+1, \ldots, b\}, \\ 0 & \text{otherwise.} \end{cases}$
- CDF: $F_X(x) = \begin{cases} 0 & \text{if } x < a\omega, \\ (\left\lfloor \frac{x}{\omega} \right\rfloor - a + 1)(b-a+1)^{-1} & \text{if } a\omega \le x \le b\omega, \\ 1 & \text{if } x > b\omega. \end{cases}$
- mean: $\omega\frac{b+a}{2}$ (average delay)

Fig. 5 shows for example the pmf and CDF of $X \sim \text{Unif}(2, 5, \omega)$ and Fig. 6 the DTP representations of different uniform distribution examples.

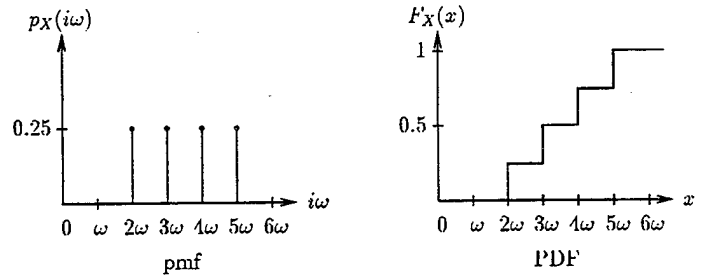


Figure 5: $X \sim \text{Unif}(2, 5, \omega)$.

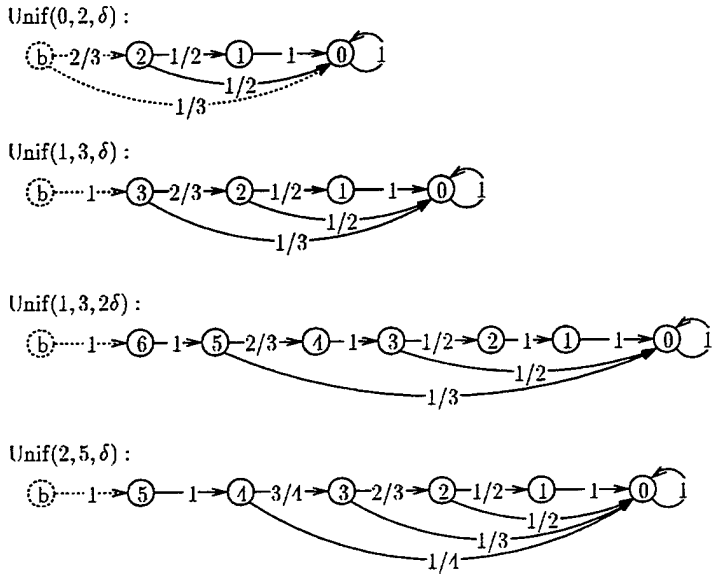


Figure 6: DTP representations of uniform distributions.

4 The DDSPN Formalism

Basic definitions of the DDSPN formalism and the specification of race policies are explained in Sections 4.1 and 4.2, respectively. Sections 4.3 and 4.4 examine the DDSPN state space and introduce the concept of well-defined DDSPNs, which is needed for the correct generation of the underlying stochastic process of a DDSPN model. Finally, Section 4.5 proposes an algorithm for the reduced reachability graph generation of a well-defined DDSPN from which the underlying stochastic process can be derived and numerically analyzed.

4.1 Basic Definitions

Informally, a DDSPN is obtained by associating a discrete time random delay, a DTP, to each PN transition. A state s of a DDSPN consists of two discrete components, the marking μ and the

vector ϕ containing the phase for each transition:

$$s = (\mu, \phi) \in \mathbb{N}^{|P|} \times \mathbb{N}^{|T|}.$$

Each entry ϕ_t of ϕ represents the current phase of the DTP associated to transition t.

Definition 4.1 Formally, a DDSPN is a tuple

$$\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}, \phi^{[0]}, \Phi, G, F, \phi^{[0]}, \not\succ, C, w\right)$$

where:

- $\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}\right)$ defines an extended PN as introduced in Section 2.
- $\forall t \in T, \Phi_t \subset \mathbb{N}$ is the finite set of phases of the DTP associated to transition t.
- $\forall \mu \in \mathcal{S}, \forall t \in T, \forall i, j \in \Phi_t, G_t(\mu, i, j)$ is the probability that the phase of transition t changes from i to j in marking μ at the end of one time-step δ. Hence, $\sum_{j \in \Phi_t} G_t(\cdot, i, j) = 1$. G_t specifies the one-step transition probability matrix of the DTP of an enabled transition t in isolation. The phase of a disabled transition does not change in isolation: $G_t(\mu, i, i) = 1$ if $t \notin \mathcal{E}(\mu)$.

  All combinations of possible new phases for all enabled transitions must be considered when ϕ is changed at the end of a step of length δ. This leads to the construction of the set $\mathcal{G}(\mu, \phi)$, such that $\forall \phi' \in \mathcal{G}(\mu, \phi)$, ϕ' is a possible combination of phases for all transitions:

$$\mathcal{G}(\mu, \phi) = \times_{t \in T} \Phi_t^G \quad \text{where} \quad \Phi_t^G = \bigcup_{\phi'_t : G_t(\mu, \phi_t, \phi'_t) > 0} \{\phi'_t\}.$$

- $\forall \mu \in \mathcal{S}, \forall t \in \mathcal{E}(\mu), \forall u \in T, \forall i, j \in \Phi_u, F_{t,u}(\mu, i, j)$ is the probability that the phase of transition u changes from i to j when transition t fires in marking μ.

  F is used for the specification of *race policies* (see Section 4.2) for transitions.

  Again, all combinations of possible new phases for all transitions need to be considered when ϕ is changed by the firing of t in μ leading to the construction of the set $\mathcal{F}(t, \mu, \phi)$, such that $\forall \phi' \in \mathcal{F}(t, \mu, \phi)$, ϕ' is a possible combination of phases for all transitions:

$$\mathcal{F}(t, \mu, \phi) = \times_{u \in T} \Phi_u^F \quad \text{where} \quad \Phi_u^F = \bigcup_{\phi'_u : F_{t,u}(\mu, \phi_u, \phi'_u) > 0} \{\phi'_u\}.$$

- $\forall t \in T, \phi_t^{[0]} \in \Phi_t$ is the initial phase of transition t at time 0.
- $\not\succ \subset T \times T$ is an acyclic post-selection priority relation.

- $C \subset 2^T$ is a partition of T into locally defined weight classes: $\forall C_x, C_y \in C, C_x \neq C_y \Rightarrow C_x \cap C_y = \emptyset$ and $\bigcup_{C_x \in C} C_x = T$. Let C_t be the local weight class containing transition $t \in T$. By setting $C_x = T$, we can model a global weight definition as in [5].
- $\forall \mu \in \mathcal{S}, \forall t \in \mathcal{E}(\mu), \forall S \subseteq C_t \cap \mathcal{E}(\mu), w_{t|S}(\mu) \in \mathbb{R}^+$ is the firing weight for t in marking μ when S is the set of *candidates* to fire in the same weight class as t. See the following description for the definition of a candidate.

□

In a DDSPN, a transition may only fire in a state where it is a *candidate*. For this reason, the enabling rule of Section 2 needs to be extended by the following definition.

Definition 4.2 A transition $t \in T$ is a *candidate* (to fire) in state $s = (\mu, \phi)$ iff it is enabled, its phase is zero, and no other candidate has post-selection priority over it (this is well defined because $\succ$ is acyclic):

1. $t \in \mathcal{E}(\mu) \ \wedge$
2. $\phi_t = 0 \ \wedge$
3. $\forall u \in T, u \not\succ t \vee u$ is not a candidate in s.

□

Moreover, the firing rule of Section 2 is extended from markings to states for DDSPNs. Let $\mathcal{C}(s)$ be the set of *candidates* in state $s = (\mu, \phi)$. Then, the probability that transition $t \in \mathcal{C}(s)$ is chosen to fire, given that one of the transitions in its weight class C_t fires, is

$$\hat{w}_{t|\mathcal{C}(s)\cap C_t}(\mu) = \frac{w_{t|\mathcal{C}(s)\cap C_t}(\mu)}{\sum_{u \in \mathcal{C}(s)\cap C_t} w_{u|\mathcal{C}(s)\cap C_t}(\mu)}$$

Note, that in DDSPNs firing probabilities are only defined among transitions belonging to the same weight class.

4.2 Race Policies

A candidate transition $t \in \mathcal{C}(s)$ may fire in a state $s = (\mu, \phi)$ leading to the new marking $\mu' = \mathcal{M}(t, \mu)$. Dynamic *race policies* [1] can be then expressed for a transition $u \in T$, where $u \neq t$, according to $F_{t,u}(\mu, \cdot, \cdot)$. This means that, depending on which transition t fired, one of the following

three *race policies* is applied to u which may cause its phase $\phi_u \in \Phi_u$ to (re)sample a random deviate $\phi'_u \in \Phi_u$ from the distribution $F_{t,u}(\mu, \cdot, \cdot)$:

R-R, race with *resampling*:

The phase of u is always resampled

$$\forall \phi'_u \in \Phi_u, F_{t,u}(\mu, \phi_u, \phi'_u) = \begin{cases} G_u(\mu, b, \phi'_u) & \text{if } u \in \mathcal{E}(\mu'), \\ 1 & \text{if } u \notin \mathcal{E}(\mu') \wedge \phi'_u = \bullet, \\ 0 & \text{otherwise.} \end{cases}$$

The resampling policy is always used when $u = t$ or when $\phi_u = \bullet$. In all other cases, it can still be used, or one of the following two policies can be used instead.

R-A, race with *age memory*:

The phase of u is not changed by the firing of t

$$\forall \phi'_u \in \Phi_u, F_{t,u}(\mu, \phi_u, \phi'_u) = \begin{cases} 1 & \text{if } \phi'_u = \phi_u, \\ 0 & \text{otherwise.} \end{cases}$$

R-E, race with *enabling memory*:

The phase of u is only resampled if u becomes disabled by the firing of t

$$\forall \phi'_u \in \Phi_u, F_{t,u}(\mu, \phi_u, \phi'_u) = \begin{cases} 1 & \text{if } u \in \mathcal{E}(\mu') \wedge \phi'_u = \phi_u, \\ 1 & \text{if } u \notin \mathcal{E}(\mu') \wedge \phi'_u = \bullet, \\ 0 & \text{otherwise.} \end{cases}$$

The approach just described allows different race policies to be applied to a transition $u \in T$ depending on which transition $t \in T$ fires. Thus, it extends the modeling power by generalizing the definition of [1], where a transition may have only a single race policy for all transition firings.

4.3 The DDSPN State Space

The underlying stochastic process of a DDSPN is a DTMC $\{(\mu^{[k]}, \phi^{[k]}) | k \in \mathbb{N}\}$ with state space $\mathcal{S} \subseteq \mathbb{N}^{|P|} \times \mathbb{N}^{|T|}$. The time-step of the DTMC is given by δ, such that $(\mu^{[k]}, \phi^{[k]}) \in \mathcal{S}$ is a DDSPN state at step k at time $k\delta$.

We adopt the terminology of [2] and call a state s *tangible* if its sojourn time is greater than zero, $\mathcal{C}(s) = \emptyset$, *vanishing* otherwise. Consequently, $\mathcal{S}$ consists only of tangible states.

Consider a tangible state $s^{[k]} = (\mu^{[k]}, \phi^{[k]})$ at time step k. At the next time step $k+1$, the new tangible state $s^{[k+1]}$ is obtained by first advancing the phases of all enabled transitions in $\phi^{[k]}$, then

by subsequently traversing vanishing states created by the possible firing of a sequence of one or more candidate transitions. A more detailed definition of a state at time step $k+1$ is then given by the following:

- Let the new tangible state reached after any firings occurring at time step $k+1$ be $s^{[k+1]}$.
- Let $s^{[k+1]0} = (\mu^{[k+1]0}, \phi^{[k+1]0})$ denote the first state reached from $s^{[k]}$ where
  - no firing occurred: $\mu^{[k+1]0} = \mu^{[k]}$ and
  - the time is advanced: $\phi^{[k+1]0} \in \mathcal{G}(\mu^{[k]}, \phi^{[k]})$.
- Let $s^{[k+1]i} = (\mu^{[k+1]i}, \phi^{[k+1]i})$, $i \in \{1, 2, ..., n\}$ denote the i-th state entered after the firing of a transition $t^i \in \mathcal{C}(s^{[k+1]i-1})$, such that
  - t^i fires: $\mu^{[k+1]i} = \mathcal{M}(t^i, \mu^{[k+1]i-1})$ and
  - the race policies are applied: $\phi^{[k+1]i} \in \mathcal{F}(t^i, \mu^{[k+1]i-1}, \phi^{[k+1]i-1})$.

  After n possible firings in n vanishing states $s^{[k+1]i}$, $i = 0, ..., n-1$, we define the first reachable tangible state to be $s^{[k+1]} \stackrel{\text{def}}{=} s^{[k+1]n}$. Note that $s^{[k+1]0} = s^{[k+1]}$ if $\mathcal{C}(s^{[k+1]0}) = \emptyset$, that is, if no firing occurs.

The previous definition describes a single state sequence $s^* = (s^{[k+1]i} | i \in \{0, 1, ..., n\})$ of states leading from $s^{[k]}$ to $s^{[k+1]}$. For better readability, let $s^{[k]} = s$ and $s^{[k+1]} = \tilde{s}$. Then, the set $S_{s,\tilde{s}}$ of all state sequences from $s = (\mu, \phi)$ to all possible $\tilde{s} = (\tilde{\mu}, \tilde{\phi})$ is given by

$$S_{s,\tilde{s}} = \Big\{ s^* = \Big(\tilde{s}^0 = (\tilde{\mu}^0, \tilde{\phi}^0), \tilde{s}^1 = (\tilde{\mu}^1, \tilde{\phi}^1), \ldots, \tilde{s}^n = (\tilde{\mu}^n, \tilde{\phi}^n) \Big) \;\Big|\; \forall \tilde{\phi}^0 \in \mathcal{G}(\mu, \phi), \quad \tilde{s}^0 = (\mu,\ \tilde{\phi}^0), \quad \forall i \in \{1, 2, ..., n\},\ \forall t^i \in \mathcal{C}(\tilde{s}^{i-1}),\ \forall \tilde{\phi}^i \in \mathcal{F}(t^i, \tilde{\mu}^{i-1}, \tilde{\phi}^{i-1}), \quad \tilde{s}^i = (\mathcal{M}(t^i, \tilde{\mu}^{i-1}),\ \tilde{\phi}^i) \Big\}.$$

The probability of a single state sequence $s^* \in S_{s,\tilde{s}}$ is then given by

$$Pr\{s^* | s^* \in S_{s,\tilde{s}} \wedge \tilde{s} \in s^*\} = g \cdot \prod_{i=1}^{n} \left(f^i \cdot F^i \right) \quad \text{where}$$

$$g = \prod_{t \in T} G_t(\mu, \phi_t, \tilde{\phi}_t^0)$$

is the probability for a single combination of phases $\tilde{\phi}^0 \in \mathcal{G}(\mu, \phi)$ and where for a transition $t^i \in \mathcal{C}(\tilde{s}^{i-1})$, such that $\tilde{\mu}^i = \mathcal{M}(t^i, \tilde{\mu}^{i-1})$

$$f^i = \hat{w}_{t^i | \mathcal{C}(\tilde{s}^{i-1}) \cap C_{t^i}}(\tilde{\mu}^{i-1})$$

is the firing probability and

$$F^i = \prod_{u \in T} F_{t^i, u}(\tilde{\mu}^{i-1}, \tilde{\phi}_u^{i-1}, \tilde{\phi}_u^i)$$

is the probability for a single combination of $\tilde{\phi}^i \in \mathcal{F}(t^i, \tilde{\mu}^{i-1}, \tilde{\phi}^{i-1})$. Fig. 7 shows a possible sequence of states leading from s to $\tilde{s}$ and the involved probabilities.

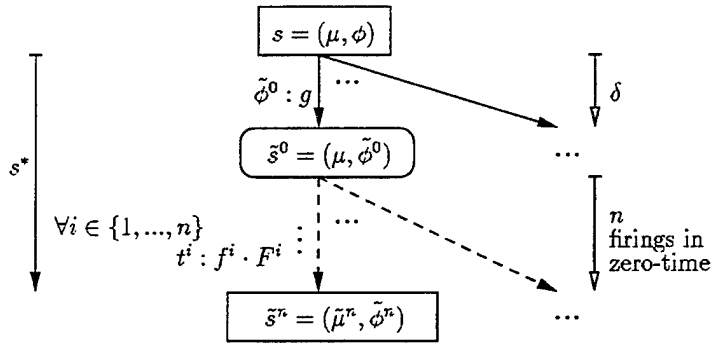


Figure 7: A sequence of states s^* leading from $s = s^{[k]}$ to $\tilde{s} = s^{[k+1]}$.

4.4 Well-defined DDSPNs

The underlying stochastic process of a DDSPN introduced in Section 4.3 takes only the tangible state space and the sojourn time in a particular state into consideration. However, for the analysis of a DDSPN, a more detailed process is needed, extending the definition of Section 4.3, to take into account the firing of transition sequences leading from one tangible state to another.

Definition 4.3 The underlying stochastic process for a DDSPN, or *basic process*, is $\{(\sigma^{[k]}, s^{[k]}) | k \in \mathbb{N}\}$, where, for $k > 0$, $\sigma^{[k]} = (t^1, ..., t^n) \in T^*$ is the k-th sequence of $n \in \mathbb{N}$ transitions to fire, at time $k\delta$, beginning from state $s^{[k]0}$ and reaching state $s^{[k]n} = s^{[k]}$, such that $s^{[k]i-1} \xrightarrow{t^i} s^{[k]i}$ for $i = 1, .., n$. ($\sigma^{[0]} = NULL$ and $s^{[k]0}$ is obtained from $s^{[k-1]}$ by advancing the time from $(k-1)\delta$ to $k\delta$). □

Informally, conflicts and confusions can arise in the context of contemporary firing attempts of PN transitions, which need to be sequentialized, and where different sequences of (formerly contemporary) transition firings lead to different undefined stochastic outcomes. A DDSPN is free of conflicts and confusions if it is *well-defined*, a precondition for its analysis. The general approach of well-defined SPNs has been first introduced in [8] and we now adapt it to give a formal definition of *well-defined* DDSPNs:

Definition 4.4 A DDSPN is *well-defined* if its basic process is completely defined, that is, if

$$\forall k \in \mathbb{N}, \forall \sigma \in T^*, \forall s \in \mathcal{S}, \ \Pr\{\sigma^{[k]} = \sigma, s^{[k]} = s\}$$

is completely determined by the elements of the DDSPN. □

In practice, we are normally interested in stochastic reward processes derived from the basic process. Without going into too much detail (see [7] for a discussion of the use of reward rates and impulses to define measures of interest), we give the following:

Definition 4.5 A stochastic process $\{Y^{[k]} \in \mathbb{R} \mid k \in \mathbb{N}^+\}$ is a *reward process* derived from the basic process through the reward structure (ρ, r) if it is defined as:

$$Y^{[k]} = \sum_{0<j\leq k} \left(\rho(\mu^{[j-1]}) \cdot \delta + \sum_{t^i \in \sigma[j]} r_{t^i}(\mu^{[j]i-1}) \right)$$

where the reward rates $\rho : \mathbb{N}^{|P|} \to \mathbb{R}$ describe the rate at which reward is accumulated in a particular marking and the reward impulses $r : (T \times \mathbb{N}^{|P|}) \to \mathbb{R}$ describe the impulse accumulated when a particular transition is fired in a particular marking. □

It is then possible for the reward process to be well-defined, even when the basic process is not. Hence we need a further:

Definition 4.6 A DDSPN is *well-defined with respect to a reward structure* (ρ, r) if

$$\Pr\{Y^{[k]} = y\}$$

is completely determined by the elements of the DDSPN, where $\{Y^{[k]} \in \mathbb{R} \mid k \in \mathbb{N}^+\}$ is the reward process defined by applying the reward structure (ρ, r) to the basic process of the DDSPN. □

Corollary 4.1 A well-defined DDSPN is well-defined with respect to any reward structure. □

The concept of well-defined SPNs and the corresponding test algorithm have been extensively discussed in [8] where more details and examples can be found.

4.5 Reduced Reachability Graph Generation

In this section we will propose an algorithm for the construction of the (finite) reduced reachability graph (RRG) and for the calculation of the impulse rewards of a well-defined DDSPN. The algorithm also tests whether the DDSPN is well-defined. The overhead for this test is small, because it is based on the state space of the RRG and on the impulse reward measures. Rate rewards are not affected by conflicts and confusions, since they are calculated before any transition firing occurs. Therefore, the calculation of rate rewards is omitted, for the sake of better readability, but it can be easily included into the algorithm. Formally, the algorithm is given:

- a DDSPN $\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}, \phi^{[0]}, \Phi, G, F, \phi^{[0]}, \not\succ, C, w\right)$, and

- a set of impulse reward functions $M = \{r^1, ..., r^{|M|}\}$,
  where $r_t^m(\mu) \in \mathbb{R}$ is the impulse reward obtained when firing transition t in marking μ according to the m-th reward structure, $1 \leq m \leq |M|$.

If the DDSPN is well-defined, the algorithm computes the underlying tangible state space $\mathcal{S}$ and all state transitions $\mathcal{P}_\mathcal{S} = \bigcup_{s \in \mathcal{S}} \mathcal{P}_s$, such that a single *path set* $\mathcal{P}_s$ contains the zero-time state transitions starting from state s. Hence, given that a tangible state $\tilde{s}$ is reachable from a state s, there is a tuple $(\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_s$ containing the corresponding state transition probability $\eta_{\tilde{s}} \in (0, 1]$ and a vector $\gamma_{\tilde{s}} = (\gamma_{\tilde{s}}^1, ..., \gamma_{\tilde{s}}^{|M|}) \in \mathbb{R}^{|M|}$, which stores the accumulated reward value $\gamma_{\tilde{s}}^m$, for every impulse reward function $r^m \in M$. A single tuple $(\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_s$ also represents the aggregated individual probabilities and accumulated impulse rewards of possible multiple paths along vanishing states leading from s to $\tilde{s}$.

The nonzero entries of the one-step transition probability matrix $\mathbf{P}$ for the underlying DTMC of a DDSPN are then given by: $\forall s, \tilde{s} \in \mathcal{S}, \forall \mathcal{P}_s \in \mathcal{P}_\mathcal{S}, \forall (\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_s : \mathbf{P}_{s,\tilde{s}} = \eta_{\tilde{s}}$. If the expected accumulated impulse rewards up to time $k\delta$, $E[Y^{[k]} \mid s^{[k]} = s]$ are known, the expected accumulated impulses up to time $(k+1)\delta$ are given by $\quad \forall k \in \mathbb{N}, \forall s, \tilde{s} \in \mathcal{S}, \forall \mathcal{P}_s \in \mathcal{P}_\mathcal{S}$:

$$E[Y^{[k+1]} \mid (s^{[k]} = s \wedge s^{[k+1]} = \tilde{s})] \;=\; E[Y^{[k]} \mid s^{[k]} = s] + \begin{cases} \gamma_{\tilde{s}} & \text{if } \exists (\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_s, \\ 0 & \text{otherwise.} \end{cases}$$

Standard numerical methods (power method, SOR) can be employed for the transient or stationary solution of the processes of interest.

If the DDSPN is not well-defined, the algorithm issues an error message and needs to be restarted after a conflict or confusion situation has been resolved by the means of priority or weight definitions. See [8] for a more detailed discussion of non-well-defined DDSPNs and their implications.

Briefly, the algorithm consists of the procedure "generateRRG" in Fig. 8, where the time is advanced in a given tangible state $s^{[k]} = s$ leading to $s^{[k+1]0} = \tilde{s}^0$, and of the procedure "traverse" in Fig. 10, where subsequent vanishing states are recursively traversed starting from $s^{[k+1]0} = \tilde{s}^0$ until tangible states $s^{[k+1]} = \tilde{s}$ are reached. Three types of parameters exist: *call by value* (in), *call by reference* (out), and *call by value-reference* (inout).

The algorithm is exercised with the call "generateRRG($\mathcal{S}$, $\mathcal{P}_\mathcal{S}$)". The set $\mathcal{S}^{next}$ contains the tangible states which have not yet been visited. It is assumed that the initial state to be visited $(\mu^{[0]}, \phi^{[0]})$ is tangible. In case of a vanishing initial state v, only the initialization of the algorithm needs to be slightly adjusted by

- first generating the set $\mathcal{S}_v^{next}$, the initial tangible states reachable from v, and then

- by storing the state transition probabilities of reaching $\mathcal{S}_v^{next}$ from v as the initial sojourn probabilities in the underlying stochastic process (a DTMC) of the DDSPN.

The initial probabilities are only relevant if a subsequent transient analysis of the DTMC is going to be performed. For the stationary analysis of the (ergodic) DTMC, it is sufficient to calculate the first reachable tangible state as the initial state; no initial probabilities are then needed.

procedure generateRRG(**out:** $\mathcal{S}$, $\mathcal{P}_\mathcal{S}$)
$\mathcal{S} = \emptyset$; $\mathcal{P}_\mathcal{S} = \emptyset$;
$\mathcal{S}^{next} = (\mu^{[0]}, \phi^{[0]})$;
while $\mathcal{S}^{next} \neq \emptyset$ **do**
  choose a state $s = (\mu, \phi)$ from $\mathcal{S}^{next}$;
  $\mathcal{S}^{next} = \mathcal{S}^{next} \setminus \{s\}$;
  $\mathcal{P}_s = \emptyset$;
  foreach $\tilde{\phi}^0 \in \mathcal{G}(\mu, \phi)$ **do**
    $\tilde{s}^0 = (\mu, \tilde{\phi}^0)$;
    $g = \prod_{t \in T} G_t(\mu, \phi_t, \tilde{\phi}_t^0)$;
    if $\mathcal{C}(\tilde{s}^0) = \emptyset$ **then** # $\tilde{s}^0$ IS TANGIBLE
      if $\tilde{s}^0 \notin \mathcal{S}$ **then**
        $\mathcal{S} = \mathcal{S} \cup \{\tilde{s}^0\}$; $\mathcal{S}^{next} = \mathcal{S}^{next} \cup \{\tilde{s}^0\}$;
      $\mathcal{P}_s^0 = \{(\gamma_{\tilde{s}^0}, g) \mid \gamma_{\tilde{s}^0} = 0\}$;
    else # $\tilde{s}^0$ IS VANISHING
      traverse($\tilde{s}^0$; $\mathcal{S}$, $\mathcal{S}^{next}$; $\mathcal{P}_{\tilde{s}^0}$);
      $\mathcal{P}_s^0 = \bigcup_{(\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_{\tilde{s}^0}} \{(\gamma_{\tilde{s}} \cdot g, \eta_{\tilde{s}} \cdot g)\}$;
    $\mathcal{P}_s = unify^-(\mathcal{P}_s, \mathcal{P}_s^0)$;
  $\mathcal{P}_\mathcal{S} = \mathcal{P}_\mathcal{S} \cup \mathcal{P}_s$;
end procedure

Figure 8: Generation of the reduced reachability graph.

Fig. 9 outlines the execution of "generateRRG". The while-loop of the procedure visits all states of $\mathcal{S}^{next}$ and calculates the set of paths $\mathcal{P}_s$, accumulated for every iteration in $\mathcal{P}_\mathcal{S}$, for each state $s \in \mathcal{S}^{next}$. The for-loop advances the time for one step δ by generating, with every iteration, the new state $\tilde{s}^0$, depending on its possible combination of next phases $\tilde{\phi}^0 \in \mathcal{G}(\mu, \phi)$ with its corresponding probability g. Moreover, for every $\tilde{s}^0$, it generates the set of paths $\mathcal{P}_s^0$ leading from s to tangible

states via $\tilde{s}^0$ and unifies them afterwards in the path set $\mathcal{P}_s$ covering all existing state transitions originating in s. Hence, if $\tilde{s}^0$ is tangible, it is added to the sets of tangible states $\mathcal{S}$ and $\mathcal{S}^{next}$, if not already there, and a single initial direct path $\mathcal{P}_s^0$ to $\tilde{s}^0$ with probability g is created with no impulse rewards, since no transition firing lead to $\tilde{s}^0$. If $\tilde{s}^0$ is vanishing, the call "traverse($\tilde{s}^0$; $\mathcal{S}$, $\mathcal{S}^{next}$; $\mathcal{P}_{\tilde{s}^0}$)" computes the path set $\mathcal{P}_{\tilde{s}^0}$ from which $\mathcal{P}_s^0$ is afterwards obtained by multiplying all impulse rewards and path probabilities of $\mathcal{P}_{\tilde{s}^0}$ with the probability g of reaching $\tilde{s}^0$ from s.

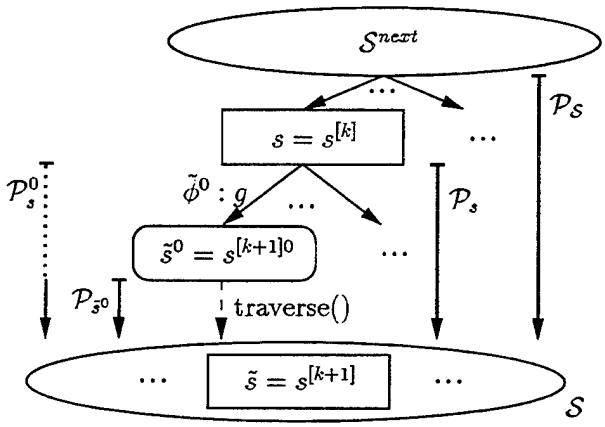


Figure 9: Execution of "generateRRG".

The function $unify^{-}(\mathcal{P}_s, \mathcal{P}'_s)$ unifies two different path sets $\mathcal{P}_s$ and $\mathcal{P}'_s$, whose origin lies in the same state s, so that multiple paths reaching the same tangible state $\tilde{s}$ are merged, guaranteeing that $\forall(\gamma_{\tilde{s}}, \eta_{\tilde{s}}), (\gamma_{\tilde{s}'}, \eta_{\tilde{s}'}) \in \mathcal{P}_s, \tilde{s} = \tilde{s}' \Rightarrow (\gamma_{\tilde{s}}, \eta_{\tilde{s}}) = (\gamma_{\tilde{s}'}, \eta_{\tilde{s}'})$. Therefore, the intersection $\mathcal{P}_s^{\cap}$ is first constructed where paths of both sets $(\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_s$ and $(\gamma'_{\tilde{s}}, \eta'_{\tilde{s}}) \in \mathcal{P}'_s$ going to the same $\tilde{s}$ are aggregated by summing the corresponding accumulated impulse rewards and path probabilities:

$$\mathcal{P}_s^{\cap} = \bigcup_{(\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_s : (\gamma'_{\tilde{s}}, \eta'_{\tilde{s}}) \in \mathcal{P}'_s} \{(\gamma_{\tilde{s}} + \gamma'_{\tilde{s}}, \eta_{\tilde{s}} + \eta'_{\tilde{s}})\}$$

Then, all paths of $\mathcal{P}_s$ and $\mathcal{P}'_s$ going to different tangible states are unified together with the intersection $\mathcal{P}_{s,\cap}$ into the set $\mathcal{P}_s^{\cup}$:

$$\mathcal{P}_s^{\cup} = \bigcup_{(\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_s \cup \mathcal{P}'_s : (\gamma'_{\tilde{s}}, \eta'_{\tilde{s}}) \notin \mathcal{P}_s^{\cap}} \{(\gamma_{\tilde{s}}, \eta_{\tilde{s}})\} \ \cup \ \mathcal{P}_s^{\cap}$$

which is also the value returned by the function.

Fig. 11 outlines the execution of "traverse". The first for-loop of the procedure in Fig. 10 partitions all candidate transitions of $\tilde{s}^{i-1}$ into sets of candidate transitions $\hat{C}_x$ belonging to the same weight class C_x. The second for-loop fires all transitions of a particular set $\hat{C}_x$, so that, with every iteration, a single candidate transition $t^i \in \hat{C}_x$ is fired in marking $\tilde{\mu}^{i-1}$ according to its firing

procedure traverse(**in:** $\tilde{s}^{i-1} \equiv (\tilde{\mu}^{i-1}, \tilde{\phi}^{i-1})$; **inout:** $\mathcal{S}$, $\mathcal{S}^{next}$; **out:** $\mathcal{P}_{\tilde{s}^{i-1}}$)

$\mathcal{P}_{\tilde{s}^{i-1}} = \emptyset$;

foreach $C_x \in C$ **do**

$\hat{C}_x = \mathcal{C}(\tilde{s}^{i-1}) \cap C_x$; $\mathcal{P}^{\hat{C}_x}_{\tilde{s}^{i-1}} = \emptyset$;

foreach $t^i \in \hat{C}_x$ **do** # FIRE SINGLE CANDIDATE

$\tilde{\mu}^i = \tilde{\mu}^{i-1} - D^-_{\bullet,t^i}(\tilde{\mu}^{i-1}) + D^+_{\bullet,t^i}(\tilde{\mu}^{i-1})$;

$f^i = \hat{w}_{t^i|\hat{C}_x}(\tilde{\mu}^{i-1})$;

foreach $\tilde{\phi}^i \in \mathcal{F}(t^i, \tilde{\mu}^{i-1}, \tilde{\phi}^{i-1})$ **do** # APPLY RACE POLICIES

$\tilde{s}^i = (\tilde{\mu}^i, \tilde{\phi}^i)$;

$F^i = \prod_{u \in T} F_{t^i,u}(\tilde{\mu}^{i-1}, \tilde{\phi}^{i-1}_u, \tilde{\phi}^i_u)$;

if $\mathcal{C}(\tilde{s}^i) = \emptyset$ **then** # $\tilde{s}^i$ IS TANGIBLE

if $\tilde{s}^i \notin \mathcal{S}$ **then**

$\mathcal{S} = \mathcal{S} \cup \{\tilde{s}^i\}$; $\mathcal{S}^{next} = \mathcal{S}^{next} \cup \{\tilde{s}^i\}$;

$\mathcal{P}^i_{\tilde{s}^{i-1}} = \{(\gamma_{\tilde{s}}, f^i F^i) \mid \forall m \in \{1, \ldots, |M|\}, \gamma_{\tilde{s}}{}^m = r^m_{t^i}(\tilde{\mu}^{i-1}) f^i F^i\}$;

else # $\tilde{s}^i$ IS VANISHING

traverse($\tilde{s}^i$; $\mathcal{S}$, $\mathcal{S}^{next}$; $\mathcal{P}_{\tilde{s}^i}$);

$\mathcal{P}^i_{\tilde{s}^{i-1}} = \bigcup_{(\gamma_{\tilde{s}}, \eta_{\tilde{s}}) \in \mathcal{P}_{\tilde{s}^i}} \{(\gamma'_{\tilde{s}}, \eta_{\tilde{s}} f^i F^i) \mid \forall m \in \{1, \ldots, |M|\},$
$\gamma'^m_{\tilde{s}} = (\gamma^m_{\tilde{s}} + r^m_{t^i}(\tilde{\mu}^{i-1}) \eta_{\tilde{s}}) f^i F^i\}$;

$\mathcal{P}^{\hat{C}_x}_{\tilde{s}^{i-1}} = unify^-(\mathcal{P}^{\hat{C}_x}_{\tilde{s}^{i-1}}, \mathcal{P}^i_{\tilde{s}^{i-1}})$;

if $\mathcal{P}_{\tilde{s}^{i-1}} = \emptyset$ **then** $\mathcal{P}_{\tilde{s}^{i-1}} = \mathcal{P}^{\hat{C}_x}_{\tilde{s}^{i-1}}$;

else if $\mathcal{P}_{\tilde{s}^{i-1}} \neq \mathcal{P}^{\hat{C}_x}_{\tilde{s}^{i-1}}$ **then** stop; # ERROR, DDSPN NOT WELL-DEFINED

end procedure

Figure 10: Traversing recursively vanishing states.

probability f^i leading to $\tilde{\mu}^i$. For each firing transition t^i the third for-loop applies the corresponding race policies to all phases of $\tilde{\phi}^{i-1}$ and generates, with every iteration, the new state $\tilde{s}^i = (\tilde{\mu}^i, \tilde{\phi}^i)$ with probability F^i depending on the possible combination of next phases $\tilde{\phi}^i \in \mathcal{F}(t^i, \tilde{\mu}^{i-1}, \tilde{\phi}^{i-1})$. Analogously to the for-loop of "generateRRG", it first generates the set of paths $\mathcal{P}^i_{\tilde{s}^{i-1}}$ leading from $\tilde{s}^{i-1}$ to tangible states via $\tilde{s}^i$, for every $\tilde{s}^i$, and then it unifies them in the path set $\mathcal{P}^{\hat{C}_x}_{\tilde{s}^{i-1}}$ covering all existing state transitions initiated by firing transitions of $\hat{C}_x$ in $\tilde{s}^{i-1}$. Again, if $\tilde{s}^i$ is tangible (terminating recursive calls), it is added to the sets of tangible states $\mathcal{S}$ and $\mathcal{S}^{next}$, if not already there. Moreover, a single initial direct path $\mathcal{P}^i_{\tilde{s}^{i-1}}$ with probability $f^i F^i$ $(Pr\{\tilde{\mu}^i\} Pr\{\tilde{\phi}^i\})$ for reaching the tangible $\tilde{s}^i$ is created together with instantaneous impulse rewards gained by the firing of t^i in

$\tilde{\mu}^{i-1}$ leading to $\tilde{s}^i$. If $\tilde{s}^i$ is vanishing, the subsequently reachable states are explored by the recursive call "traverse($\tilde{s}^i$; $\mathcal{S}$, $\mathcal{S}^{next}$; $\mathcal{P}_{\tilde{s}^i}$)" which computes the path set $\mathcal{P}_{\tilde{s}^i}$ (assuming that the vanishing reachability graph created from $\tilde{s}^i$ is acyclic, finite, and that no conflict or confusion occurred). $\mathcal{P}^i_{\tilde{s}^{i-1}}$ is then obtained from $\mathcal{P}_{\tilde{s}^i}$ by adding the instantaneous impulse rewards of t^i to $\mathcal{P}_{\tilde{s}^i}$, so that the probability of a particular path and of the accumulated impulse rewards equals to $\eta'_{\tilde{s}} = f^i F^i \eta_{\tilde{s}}$ for reaching a tangible state $\tilde{s}$ from $\tilde{s}^{i-1}$ via $\tilde{s}^i$.

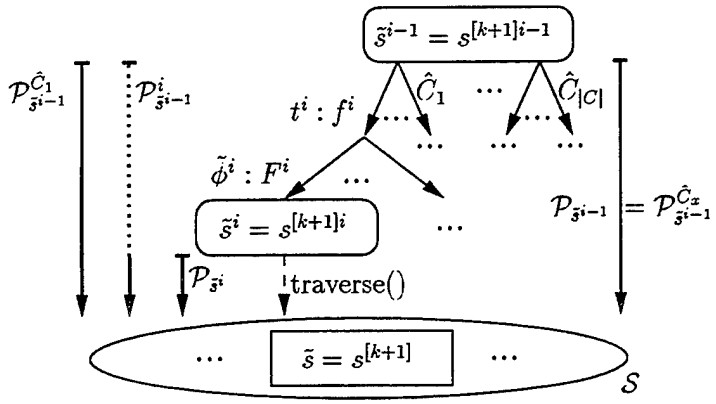


Figure 11: Execution of "traverse".

Conflicts and confusions exhibit a non-deterministic behavior which can occur in DDSPNs only in a vanishing state $\tilde{s}^{i-1}$ when multiple candidate transitions $t \in \mathcal{C}(\tilde{s}^{i-1})$ attempt to fire in zero-time leading to tangible states $\tilde{s}$ with different stochastic outcomes. Indeed, the DDSPN evolution during instants of time where there is no firing is completely determined by the assumption of a race behavior.

It is possible to resolve conflicts and confusions by employing one of the following two methods. Priorities can be defined to prevent conflicting transitions from becoming simultaneous candidates, hence from attempting to fire at the same time. The second method groups candidate transitions involved in conflicts or confusions into the same weight class C_x, so that contemporary firing attempts are resolved probabilistically by the individual firing probability f^i for each candidate $t^i \in \hat{C}_x$ where $\sum_{t^i \in \hat{C}_x} f^i = 1$. Then, candidate transitions belonging to different weight classes are free of conflicts and confusions, and they reach from all vanishing states $\tilde{s}^{i-1}$ the same tangible states with the same probabilities and with the same accumulated impulse rewards, regardless of the order in which they are fired (a necessary condition for the absence of conflicts and confusions), such that $\forall C_x \in C : \mathcal{P}^{\hat{C}_x}_{\tilde{s}^{i-1}} = \mathcal{P}_{\tilde{s}^{i-1}}$.

If a vanishing state is encountered with at least two different paths sets where $\mathcal{P}^{\hat{C}_x}_{\tilde{s}^{i-1}} \neq \mathcal{P}^{\hat{C}_y}_{\tilde{s}^{i-1}}$, the DDSPN is not well-defined with regard to the particular reward processes of interest. The

modeler must then apply one of the following actions to transitions of $\hat{C}_x$ and $\hat{C}_y$ before restarting the algorithm, to remove conflicts or confusions:

- Specify pre-selection priorities disabling a conflicting transition before the advance of time.
- Specify post-selection priorities, thus forcing a particular sequence for contemporary firing attempts.
- Merge the corresponding weight-classes of conflicting transitions and define appropriate weights for them.

5 State Space Reduction

In Section 4.5, it has been shown how the one-step transition probability matrix $\mathbf{P}$ of the underlying (finite) DTMC of a DDSPN is computed. In case of an irreducible DTMC, the stationary solution is obtained by solving the following system of linear equations with standard techniques (Gauss-Seidel, SOR): $\pi = \pi\mathbf{P}$ and $\sum_i \pi_i = 1$.

Since $\mathbf{P}$ is usually a sparse matrix, sparse storage schemes should be employed. Measures of interest are then derived from the stationary probability distribution vector π.

A considerable reduction of the state space can be achieved if it is possible to advance the phases of enabled transitions during the state space generation for more than just one time-step δ until a probabilistic split or a phase equal to zero (vanishing state) is reached. This is the case when tangible states are encountered where the DTPs of the enabled transitions have a unit-step ω which is a multiple of the basic underlying time-step δ, a condition often met by deterministic transitions. Hence, the algorithm for the RRG generation is slightly modified to test whether the next maximum x phase advancements of all enabled transitions of a tangible state of $\mathcal{S}^{next}$ have probabilities equal to one. Consider a tangible state $s^{[k]} = (\mu^{[k]}, \phi^{[k]})$ at time $k\delta$ from which the following sequence of states is initiated

$$s^* = (s^{[k+i]} = (\mu^{[k]}, \phi^{[k+i]}) \mid i \in \{1, 2, ..., x\}, x > 1),$$

so that:

$$\exists x \in \mathbb{N}^+, \forall i \in \{1, 2, ..., x\} : Pr\left\{s^{[k+i]} = (\mu^{[k]}, \phi^{[k+i]}) \middle| s^{[k+i-1]} = (\mu^{[k]}, \phi^{[k+i-1]})\right\} = 1 \wedge \mathcal{C}(s^{[k+i-1]}) = \emptyset.$$

Since no change of marking and no transition firing occurred, the states of s^* were generated solely by phase advancements of enabled transitions whose phase transition probabilities equal to one, hence,

$$\forall s^{[k+i]} = (\mu^{[k]}, \phi^{[k+i]}) \in s^*, \forall t \in \mathcal{E}(s^{[k+i-1]}) : G_t(\mu^{[k]}, \phi_i^{[k+i-1]}, \phi_i^{[k+i]}) = 1.$$

Therefore, it becomes possible to advance directly from $s^{[k]}$ to $s^{[k+x]}$ leaving out the intermediate tangible states $\{s^{[k+i]} | i \in \{1, 2, ..., x-1\}\}$ while adding their individual holding times of $h_{s^{[k+i]}} = \delta$ to the holding time of $s^{[k]}$, so that $h_{s^{[k]}} = x\delta$. If the last state of the sequence is vanishing, $s^{[k+x]0}$ is reached instead of $s^{[k+x]}$.

The underlying stochastic process is then a discrete time semi-Markov process where $\mathbf{P}$ describes an embedded DTMC. The holding times in each state are no longer equal to δ, but are given by the holding time vector h. The stationary solution can be obtained employing the following well-known method for semi-Markov processes [9]: We first solve the system of linear equations $\gamma = \gamma\mathbf{P}$ and $\sum_i \gamma_i = 1$ for the embedded stationary probabilities γ; then, we rescale γ using the holding times, $\forall s \in \mathcal{S} : \gamma'_s = \gamma_s \cdot h_s$; finally, we normalize the rescaled probabilities γ' and obtain the stationary probability distribution: $\pi = \frac{\gamma'}{\sum_i \gamma'_i}$.

In general the size of the state space depends on the size of the basic underlying time-step δ and on the number of phases of firing time distributions (DTPs) specified for the timed transitions of a DDSPN model. If embedding is used, the size of the state space depends, in addition, on the maximum possible phase advancements of all enabled transitions in tangible states.

6 Example

This section illustrates the modeling power of DDSPNs by presenting an example containing several deterministically timed activities. Consider the processing station of an automated manufacturing system where raw parts arrive at constant time intervals. A machine tool processes each raw part for a constant time period. The tool wears off and needs to be replaced after a stochastically timed delay whose value depends on the tool quality and on the material of the processed parts. The time delay for the replacement is constant. The processing station can be then characterized by a D/D/1/K queueing system where the service station (tool) is subject to stochastic failures (wearout) and deterministically timed repairs (tool replacement). Fig. 12 shows the corresponding DDSPN model. Raw parts, represented by tokens, arrive with the firing of the deterministic transition *arriving* and wait for service on place *WAIT* until the service station is empty and operable. A single token on place *IDLE SERVER* and the immediate transition *enter service* permit only one part at a time to enter the service station which consists of the place *SERVICE* and of the deterministic transition *serve*. The firing of *serve* stands for the completion of the processing of a single part. The failure and repair of the service station are represented by the geometric and deterministic transitions *failure* and *repair*, respectively.

We consider a system with $K = 50$ parts, a constant deterministic arrival rate of $\frac{1}{10s}$, and a

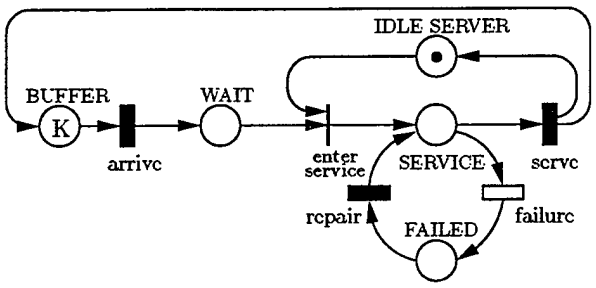
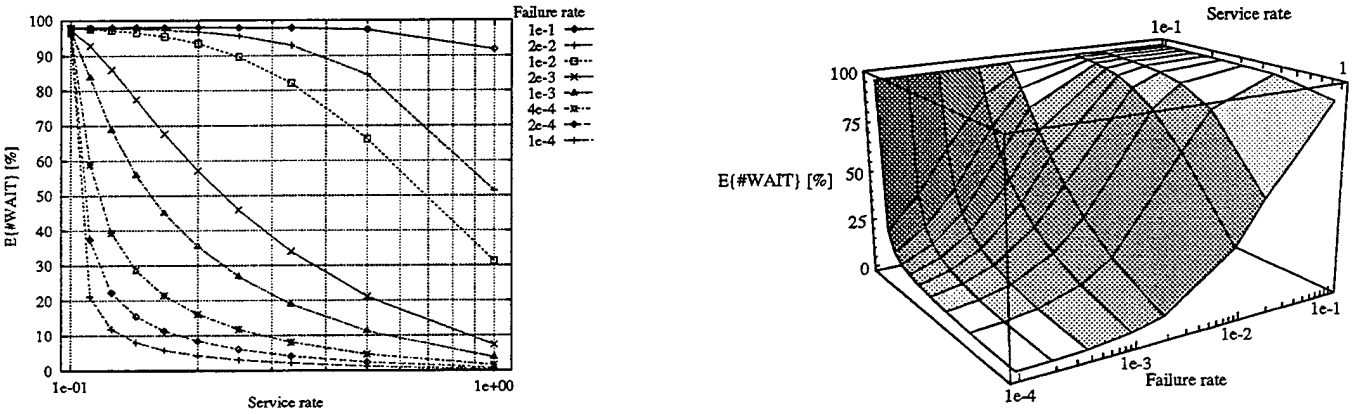


Figure 12: D/D/1/K queueing system with failure and repair.

constant deterministic repair rate of $\frac{1}{10^3 s}$. The deterministic service rate is varied from $\frac{1}{10s}$ to $\frac{1}{1s}$ and the geometric failure rate is varied from $\frac{1}{10s}$ to $\frac{1}{10^4 s}$. The basic underlying time-step of the model equals to $1s$.

The measure of interest of the stationary solution is the average number of waiting raw parts $E\{\#WAIT\}$ on place *WAIT* depending on the varying service and failure rates. The goal of our performance evaluation is to determine which minimum performance of the server, in terms of speed (service rate) and dependability (failure rate), suffices to achieve a desired average percentage of waiting raw parts. Fig. 13 shows the corresponding curves, where $E\{\#WAIT\}$ is plotted vs. the firing rates of transition *serve* and transition *failure*.



a) 2D

b) 3D

Figure 13: Mean number of waiting raw parts (in rate and failure rate.

The state space of the DDSPN consists of 101 tangible markings. Depending on the deterministic service rate 5,930 up to 125,153 tangible states have been generated. However, employing the embedding technique for the stationary solution of this particular model leads to a state space reduction of 86.4

7 Conclusion

The results of [14],[5], and [8] have been combined introducing the DDSPN formalism where deterministic and stochastic firing times of transitions can be mixed without structural restrictions while providing integrated automatic conflict and confusion detection on a discrete time scale.

A new solution method combining [5] and [8] and a previously not available algorithm for mapping a DDSPN onto its underlying stochastic process have been presented from which a direct implementation can follow. Thus, a new practical formalism in the field of performance evaluation has been enabled with new features based on discrete time as demonstrated for a typical queueing application example.

Considerable state space reduction can be achieved for a given DDSPN model by carefully choosing timing parameters and, more importantly, by means of embedding. Even so, the DDSPN formalism still leads to a large state space due to the additional phase components in the state.

References

[1] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analyis of Stochastic Petri Nets. *IEEE Trans. on Softw. Eng.*, 15(7):832–846, July 1989.

[2] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of Generalized Stochastic Petri Nets for the performance analysis of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(1):93–122, May 1984.

[3] M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In G. Rozenberg, editor, *Advances in Petri Nets 1987, Lecture Notes in Computer Science*, volume 266, pages 132–145. Springer, 1987.

[4] G. Ciardo. Petri nets with marking-dependent arc multiplicity: Properties and analysis. In R. Valette, editor, *Application and Theory of Petri Nets 1994, Lecture Notes in Computer Science 815 (Proc. 15th Int. Conf. on Applications and Theory of Petri Nets, Zaragoza, Spain)*, pages 179–198. Springer, June 1994.

[5] G. Ciardo. Discrete-time Markovian Stochastic Petri Nets. In W. J. Stewart, editor, *Numerical Solution of Markov Chains '95*, pages 339–358, Raleigh, NC, USA, Jan. 1995.

[6] G. Ciardo. Advances in compositional approaches based on kronecker algebra: Application to the study of manufacturing systems. In *3rd Int. Workshop on Performability Modeling of Computer and Communication Systems (PMCCS3)*, pages 61–65, Bloomingdale, Illinois, USA, Sep. 1996.

[7] G. Ciardo, A. Blakemore, P. Chimento, J. Muppala, and K. Trivedi. Automated generation and analysis of Markov reward models using Stochastic Reward Nets. In C. Meyer and R. J. Plemmons, editors, *Linear Algebra, Markov Chains, and Queueing Models*, volume 48 of *IMA Volumes in Mathematics and its Applications*, pages 145–191. Springer, 1993.

[8] G. Ciardo and R. Zijal. Well-defined Stochastic Petri Nets. In *Proc. 4th Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'96)*, pages 278–284, San Jose, CA, USA, Feb. 1996. IEEE Computer Society Press.

[9] E. Çinlar. *Introduction to Stochastic Processes.* Prentice-Hall, 1975.

[10] R. German. *Analysis of Stochastic Petri Nets with Non-Exponentially Distributed Firing Times.* PhD thesis, Technical University of Berlin, Berlin, Germany, 1994.

[11] M. Holliday and M. Vernon. A Generalized Timed Petri Net model for performance analysis. *IEEE Trans. on Softw. Eng.*, 13(12):1297–1310, Dec. 1987.

[12] M. Molloy. Discrete time Stochastic Petri Nets. *IEEE Trans. on Softw. Eng.*, 11(4):417–423, April 1985.

[13] C. A. Petri. *Kommunikation mit Automaten.* PhD thesis, University of Bonn, Bonn, West Germany, 1962.

[14] R. Zijal. Discrete Time Deterministic and Stochastic Petri Nets. In G. Hommel, editor, *Proc. Int. Workshop - Quality of Communication-Based Systems*, pages 123–136, TU-Berlin, Germany, Sept. 1994. Kluwer.

[15] W. Zuberek. Timed Petri Nets, definitions, properties, and applications. *Microelectronics and Reliability*, 31(4):627–644, 1991.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1996	Contractor Report

4. TITLE AND SUBTITLE
DISCRETE DETERMINISTIC AND STOCHASTIC PETRI NETS

5. FUNDING NUMBERS
C NAS1-19480
WU 505-90-52-01

6. AUTHOR(S)
Robert Zijal
Gianfranco Ciardo

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Computer Applications in Science and Engineering
Mail Stop 403, NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 96-72

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-201629
ICASE Report No. 96-72

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Dennis M. Bushnell
Final Report
To be submitted to the Workshop on Measurement, Modeling, & Evaluation of Comp. & Comm. Systems.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited

Subject Category 60, 61

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Petri nets augmented with timing specifications gained a wide acceptance in the area of performance and reliability evaluation of complex systems exhibiting concurrency, synchronization, and conflicts. The state space of time-extended Petri nets is mapped onto its basic underlying stochastic process, which can be shown to be Markovian under the assumption of exponentially distributed firing times. The integration of exponentially and non-exponentially distributed timing is still one of the major problems for the analysis and was first attacked for continuous time Petri nets at the cost of structural or analytical restrictions. We propose a discrete deterministic and stochastic Petri net (DDSPN) formalism with no imposed structural or analytical restrictions where transitions can fire either in zero time or according to arbitrary firing times that can be represented as the time to absorption in a finite absorbing discrete time Markov chain (DTMC). Exponentially distributed firing times are then approximated arbitrarily well by geometric distributions. Deterministic firing times are a special case of the geometric distribution. The underlying stochastic process of a DDSPN is then also a DTMC, from which the transient and stationary solution can be obtained by standard techniques. A comprehensive algorithm and some state space reduction techniques for the analysis of DDSPNs are presented comprising the automatic detection of conflicts and confusions, which removes a major obstacle for the analysis of discrete time models.

14. SUBJECT TERMS
Deterministic and Stochastic Petri nets; discrete time; reward processes; conflicts and confusion

15. NUMBER OF PAGES
25

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		